United Electronic Industries

The High-Performance Alternative

DNx-AI-201

—

User Manual

Sequential Sampling, 16-bit, 24-channel Analog Input Layer for the PowerDNA Cube and PowerDNR RACKtangle

Release 4.6

September 2013

PN Man-DNx-AI-201-913

Contacting United Electronic Industries

Mailing Address:

27 Renmar Avenue
Walpole, MA 02081
U.S.A.

For a list of our distributors and partners in the US and around the world, please see http://www.ueidaq.com/partners/

Support:

Telephone: (508) 921-4600
Fax: (508) 668-2350

Also see the FAQs and online “Live Help” feature on our web site.

Internet Support:

Support: support@ueidaq.com
Web-Site: www.ueidaq.com
FTP Site: ftp://ftp.ueidaq.com

Product Disclaimer:

WARNING!

DO NOT USE PRODUCTS SOLD BY UNITED ELECTRONIC INDUSTRIES, INC. AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS.

Products sold by United Electronic Industries, Inc. are not authorized for use as critical components in life support devices or systems. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness. Any attempt to purchase any United Electronic Industries, Inc. product for that purpose is null and void and United Electronic Industries Inc. accepts no liability whatsoever in contract, tort, or otherwise whether or not resulting from our or our employees' negligence or failure to detect an improper purchase.

Specifications in this document are subject to change without notice. Check with UEI for current status.

Table of Contents

List of Figures

Chapter 1 Introduction

This document outlines the feature set and use of the DNR- and DNA-AI-201 layer. The AI-201 is a four-channel strain gauge input module for the PowerDNA I/O Cube (DNA-AI-201) and the PowerDNR HalfRACK, RACKtangle, and the FlatRACK chassis (DNR-AI-201).

1.1 Organization of Manual

This AI-201 User Manual is organized as follows:

- **Introduction**
  This chapter provides an overview of DNx-AI-201 Analog Input Board features, device architecture, connectivity, and logic.
- **Programming with the High-Level API**
  This chapter provides an overview of the how to create a session, configure the session, and interpret results with the Framework API.
- **Programming with the Low-Level API**
  This chapter is an overview of low-level API commands for configuring and using the AI-201 series layer.
- **Appendix A - Accessories**
  This appendix provides a list of accessories available for use with the DNx-AI-201 board.
- **Index**
  This is an alphabetical listing of the topics covered in this manual.

Manual Conventions

To help you get the most out of this manual and our products, please note that we use the following conventions:

Tips are designed to highlight quick ways to get the job done or to reveal good ideas you might not discover on your own.

NOTE: Notes alert you to important information.

CAUTION! *Caution advises you of precautions to take to avoid injury, data loss, and damage to your boards or a system crash.*

Text formatted in **bold** typeface generally represents text that should be entered verbatim. For instance, it can represent a command, as in the following example: “You can instruct users how to run setup using a command such as **setup.exe**.”

Text formatted in `fixed` typeface generally represents source code or other text that should be entered verbadim into the source code, initialization, or other file.

Examples of Manual Conventions

Before plugging any I/O connector into the Cube or RACKtangle, be sure to remove power from all field wiring. Failure to do so may cause severe damage to the equipment.

Usage of Terms

Throughout this manual, the term “Cube” refers to either a PowerDNA Cube product or to a PowerDNR RACKtangle™ rack mounted system, whichever is applicable. The term DNR is a specific reference to the RACKtangle, DNA to the PowerDNA I/O Cube, and DNx to refer to both.

1.2 The AI-201 Interface Board

The DNA/DNR-AI-201 is a 24-channel single ended, 12-channel differential A/D board. The DNA-boards are compatible with all of our popular "CUBE" series chassis while the DNR-series boards are used in the RACKtangle® I/O series chassis. Both versions provide identical electrical specifications and performance including 16-bit resolution with a maximum input range of ±15V. The board is fully isolated from the PowerDNA cube and is the ideal A/D board for a wide variety of high speed, high resolution data acquisition (DAQ) and control applications.

1.3 Features

The AI-201 layer has the following features:

- 24 single-ended/12 differential inputs
- ±15V input range
- “DNx-AI-201-16” version offers gains of 1/10/100/1000, per-channel selectable, maximum speed is 16kS/sec on multiple channels
- “DNx-AI-201-100” and -801 version offer gains of 1/2/5/10, per-channel selectable, maximum speed is 100kS/sec on multiple channels
- Completely configurable per-channel settling time delay with better than 1µS resolution
- 16-bit resolution, no missing codes
- ± 40V overvoltage, 2kV ESD protection on all input pins
- All analog I/O is fully opto-isolated from the rest of the system
- Additional dedicated differential channel for CJC compensation
- Auto-calibration (software initiated)
- Precision, digitally calibrated to 0.001%, 3ppm/°C on-board reference
- 3 configurable multi-purpose DIO lines (default configuration – CLKOUT/CLKIN/TRIGIN), value of the DIO lines delivered along with analog input sample
- 1k samples input FIFO with 32-bit per-sample timestamp
- Interrupt request on any position in the input or channel list FIFO
- On-board EEPROM to store configuration and calibration data
- Input ground to system ground isolation: $350V_{rms}$
- Power consumption ~ 1.6/2.2W
- Weight of 120 g or 4.24 oz for DNA-AI-217; 630 g or 22.2 oz with PPC5
- UEI Framework Software API may be used with all popular Windows programming languages and most real time operating systems such as RT Linux, RTX, or QNX and graphical applications such as LabVIEW, MATLAB, DASYLab and any application supporting ActiveX or OPC

1.4 Specification

The technical specification for the DNx-AI-201-100 board are listed in **Table 1-1**.

Table 1-1. DNx-AI-201-100 Technical Specifications

Resolution	16 bits
Number of Channels: Single-Ended Differential	 24 12
Maximum Sampling Rate	100 kS/s, aggregate
Onboard FIFO Size	512 samples
Input Range	±15V
Programmable Gains	1, 2, 5, 10 (by channel)
Input Impedance	10 MΩ
Input Bias Current	±15 nA
Input Overvoltage	±40V, 2000V ESD powered or unpowered
A/D Conversion Time	2 µs
A/D Settling Time	10µs @ G=1; 15µs @ G=2; 25µs @ G=5; 50µs @ G=10
Nonlinearity	1 LSB
System Noise	1.2 LSB
Isolation	$350V_{rms}$
Effective Number of Bits	14.8
Total Harmonic Distortion+Nonlinearity+Noise	91 dB
Channel Crosstalk	85 dB @ 1 kS/s
Power Consumption	2.0W
Physical Dimensions	3.875 x 3.875″(98 x 98 mm)
Operating Temp. (tested)	-40°C to +85°C
Operating Humidity	95%, non-condensing
Vibration *IEC 60068-2-6* *IEC 60068-2-64*	5 g, 10-500 Hz, sinusoidal 5 g (rms), 10-500 Hz, broad-band random
Shock *IEC 60068-2-27*	50 g, 3 ms half sine, 18 shocks @ 6 orientations 30 g, 11 ms half sine, 18 shocks @ 6 orientations

1.5 Device Architecture

Figure 1-1 is a block diagram of the architecture of the AI-201 layer.

The DNx-AI-201 layer is physically divided into isolated and non-isolated sections. The non-isolated part is powered from DC/DC converters located on the CPU layer. These converters provide 5V and 3.3/2.5V to power all non-isolated electronics.

The isolating DC/DC that produces 5V powers the isolated side of the layer. Two high-frequency boosters generate +18V,-18V rails that are also available on the connector (15mA maximum each). On the 201-801 this is a +2.5V,-18V.

The DNx-AI-201 employs a single successive-approximation 18-bit converter with no pipeline delay.

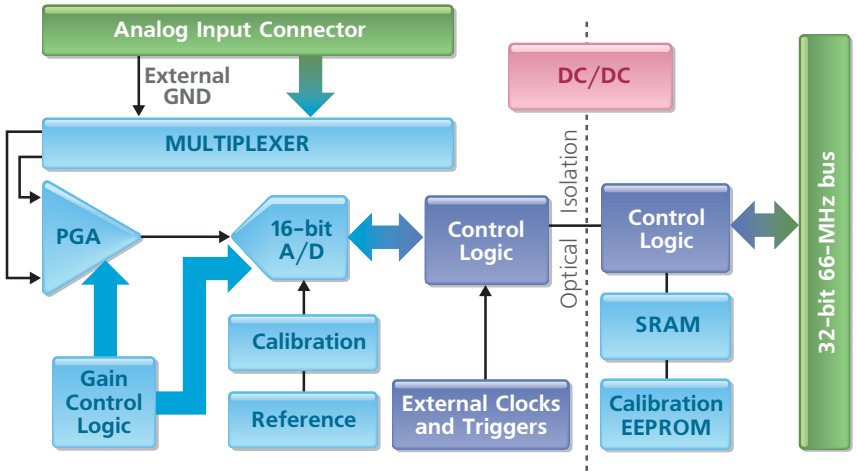


Figure 1-1. Architecture Block Diagram of the AI-224 Layer

1.6 Indicators

A photo of the DNx-AI-201 unit is illustrated below.

The front panel has two LED indicators:

- RDY: indicates that the layer is receiving power and operational.
- STS: can be set by the user using the low-level framework.

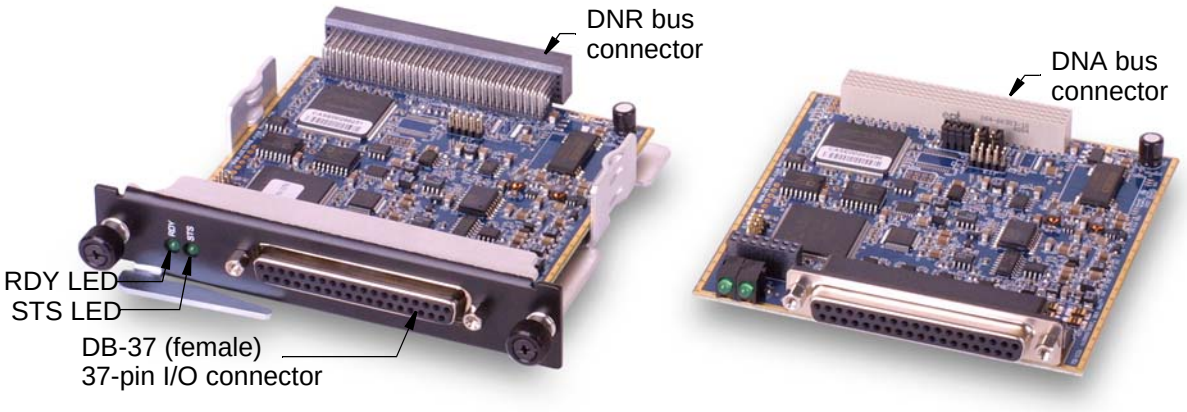


Figure 1-2. The DNA-AI-201 Analog-Input Layer

1.7 Layer Connectors and Wiring

Figure 1-3 below illustrates the pinout of the AI-201. Like other layers, it uses a B-size 37-pin D-sub connector for all I/O connections.

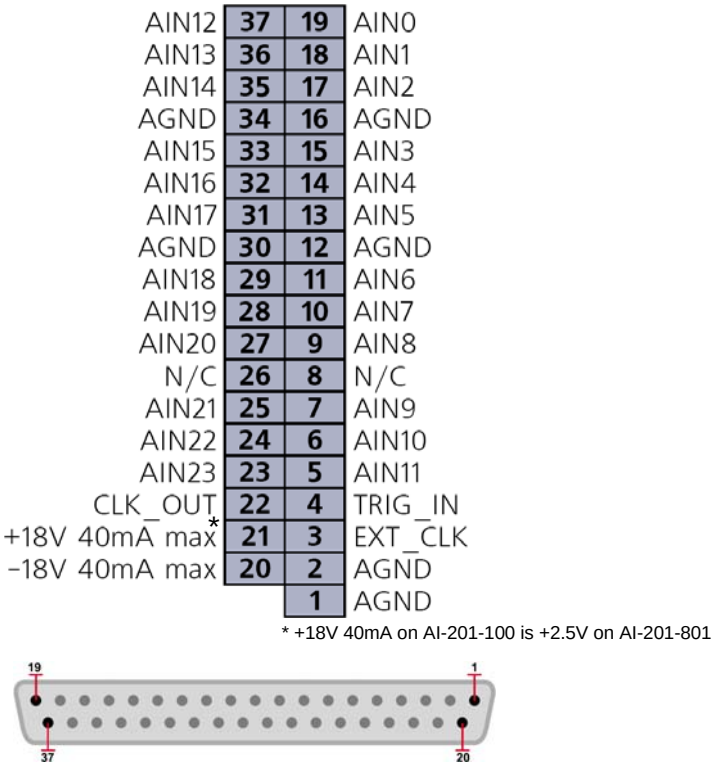


Figure 1-3. Pinout Diagram of the DNx-AI-201

The following signals are located at the connector:

- ***AIn0 – AIn23*** input channels. If a channel is set to operate as single-ended, its voltage is measured between the selected channel line and the analog ground (*AGND*) line. If Channel X is used as a differential measurement, its inputs are connected between Channel *AIn(X)* and *AIn(X+12).* For example, in differential mode, channel *AIn0* pairs with *AIn12*, *AIn1* with *AIn13*, etc.
- ***AGND*** – analog ground of the layer, isolated from system ground
- ***TCPOS* and *TCNEG*** – dedicated channels to connect a thermocouple compensation device. Use signal and return to measure in differential mode or just *TCPOS* to measure in single-ended mode relative to *AGND*. *TCPOS* should then be programmed as channel *AIn24*.
- ***CLKOUT*** – this line, by default, is an output and is used as an external channel list clock to synchronize multiple PowerDNA cubes. This line can also be used as a bi-directional general-purpose DIO.
- ***TRIGIN*** – by default, this is an input that provides an external trigger signal to the layer. This line can also be used as bi-directional general-purpose DIO.
- ***EXTCLK***- by default, this is an input that provides external an CV or CL clock to the layer logic. This line can also be used as a bi-directional general-purpose DIO.

- ***+18V, -18V*** lines on the AI-201-100 and AI-201-16 provide isolated voltage generated on the layer to power external sensors. As maximum current is limited to 15mA each, we suggest using large current-limiting resistors when using this source to power strain gages or RTDs[1]. On the AI-201-801 the reference is +2.5V, -18V instead.

1.7.1 Analog Input Ground Connections

To avoid errors caused by common mode voltages on analog inputs, follow the recommended grounding guidelines in **Figure 1-4** below.

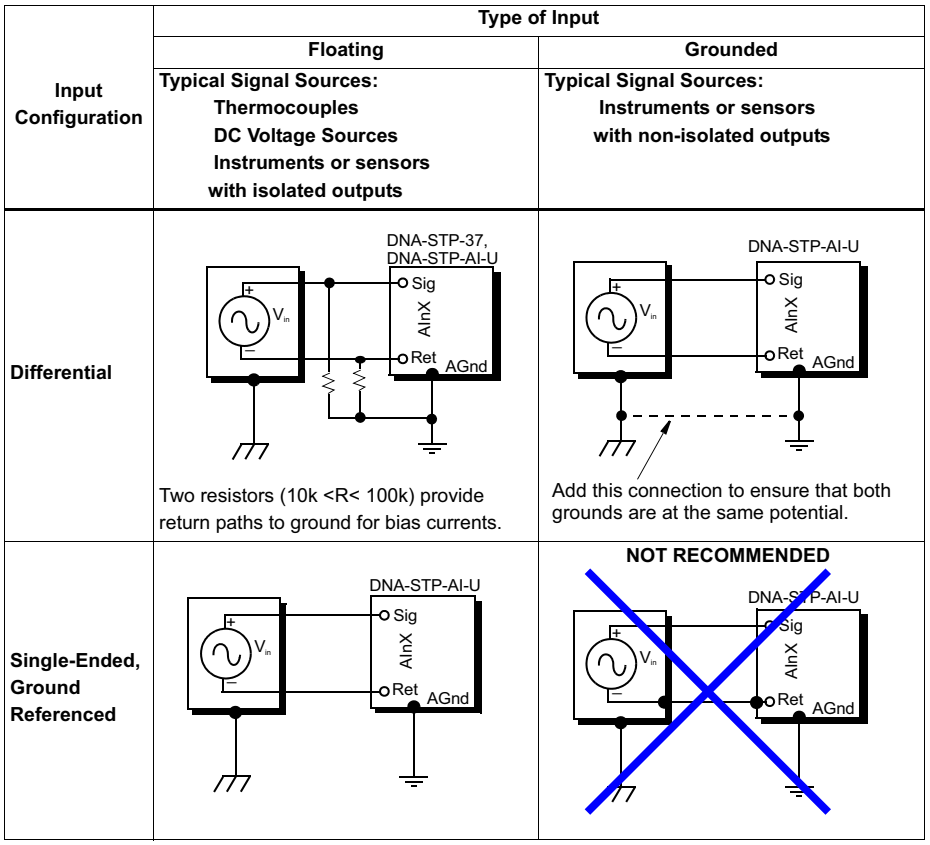


Figure 1-4. Recommended Ground Connections for Analog Inputs

Because all analog input channels in AI-201/202/207/208/225 layers are isolated as a group, you can connect layer AGND to the ground of the signal source and eliminate the resistors shown in **Figure 1-4** for floating differential input signals.

1. RTD is a Resistance Temperature Detector - a resistor that changes resistance with its temperature, sometimes called Platinum Resistance Thermometers (PRT) because most of them are fabricated of platinum wire or thin film.

1.8 Layer Capabilities

A layer is capable of acquiring analog input voltage in the ±15V range with gains of 1, 2, 5, 10 (DNx-AI-201-100/-801) or 1, 10,100, 1000 (DNx-AI-201-16).

A layer is capable of generating its own CL and CV clocks and trigger and deriving them from either local external lines from its connector or from the SYNCx bus.

You can connect a signal source to the layer using either differential or single-ended wiring. Because of this, the ground plane is isolated from the system ground; single ended and pseudo-differential wiring use effectively the same wiring.

Table 1-2. Gains

Card	Gain	Range	Settling Time to 16-bit Resolution	Noise, LSB	Resolution, Noise Limited
DNA-AI-201-16	1	± 15V	62.5µs	0.81	457µV
	10	±1.5V	100µs	1.05	45µV
	100	±150mV	800µs	1.58	40µV
	1000	± 15mV	2.1ms	4.32	30µV
DNA-AI-201-100 & DNA-AI-201-801	1	±15V	10µs	1.04	457µV
	2	±7.5V	15µs	1.28	228µV
	5	±3V	25µs	2.20	91µV
	10	±1.5V	50µs	3.11	45µV

1.8.1 Single-Ended and Pseudo-Differential

An AI-201 layer operating in single-ended mode digitizes across as many as 24 channels. For single-ended inputs, you connect one wire from each signal source to the High input of the input amplifier to the data-acquisition system. All signals share a common return path connected to analog ground (AGND). You should connect this common return path both to a ground near the signal source and also to the ground on the PC, which forces it to be the same level as the signal ground. This ground signal, however, is typically referenced to a remote source and because it is separated from the system ground, it can float at a different level. The maximum potential difference between common ground and PC ground should never exceed 350V. Because the AGND line in a pseudo-differential setup is not connected to the computer ground, however, it is not subject to the associated digital noise within the system.

1.8.2 Differential

An AI-201 layer operating in differential mode digitizes as many as 12 channels. Each channel uses two lines on the input amplifier of the data-acquisition system — you connect one lead from the signal source to the channel's High input (the positive input of the amp) and connect the other signal lead to the channel's Low input (the amp's negative input). Each signal floats at its own level without any reference to ground or other inputs.

When working with a DNx-AI-201 layer in differential mode, AIn0 and AIn12 form the High and Low inputs of differential-input Ch 0; next, for differential-input Ch 1, use AIn1 and AIn13; follow this pattern for all twelve differential-input pairs.

Two high-impedance amplifiers monitor the voltage between the inputs and the PC ground. A third amplifier measures the difference between the Positive and Negative inputs, eliminating any voltage common to both wires. This method eliminates problems that can arise with a single-ended system because this configuration attenuates noise common to both channel inputs (common-mode noise). Thus, it's wise to use twisted-pair cable to bring signals to the data-acquisition layer because that setup ensures that any noise generated along the wiring path is the same for each line, and the amplifier subtracts this noise.

Although using differential inputs cuts in half the number of channels you can read with a given layer compared to single-ended or pseudo-differential setups, there are several cases where you are well advised to use differential inputs, such as:

- When signal leads are over a few meters in length.The instrumentation amp can eliminate the effect of noise pickup from signal leads and eliminate the possibility of ground differentials.
- When signals are less than approximately 100 mV. Such low-level signals can be easily overwhelmed by noise and ground differentials that only the differential mode can remove.
- When the signals are from high-impedance sensors such as strain gauges. Their high impedances can lead to higher common-mode voltages, which differential inputs are able to remove.

1.8.3 Thermocouple Measurement

Although the DNx-AI-201 is capable of performing thermocouple measurement, such a task is best reserved for layers with higher resolution and better noise rejection, such as the DNx-AI-225 or the DNx-AI-207.

The higher the speed of measurement, the more noise can be expected. The following table shows test results for noise for the AI-201 — when used in conjunction with the STP-AI-U terminal, in differential mode, with high gains.

Table 1-3. Gain Performance of DNx-AI-201 Layer with Various Thermocouple Types

Thermocouple Type	Gain 1000 Performance			Gain 100 Performance		
	Temperature Range	P-P Noise	100 Points RMS Noise	Temperature Range	P-p Noise	100 Points RMS Noise
B	0-1810	3	1	Full range of thermocouple	8	2
C	0-800	1	0.3		2	1
E	-290 to +220	0.5	0.15		1	0.35
J	-210 to +270	0.5	0.15		1	0.35
K	-26 to +360	0.5	0.2		1.5	0.5
N	-270 to +450	1	0.3		2	1
R	-50 to +1320	2	0.8		5	1.8
S	-50 to +1450	2	0.8		6	2

Additional Factors to Consider:

- Channel-channel offset due to the multiplexed architecture is ±10uV typical
- Open TC detection circuitry on the DNA-STP-AI-U adds ~ 35uV (±5uV) offset on all channels (may be compensated by using CJC temperature sensor calibration)
- CJC sensor is calibrated to better than 0.2 ºC accuracy at room temperature, and stays within 0.4 ºC accuracy within the -20..+75ºC temperature range

1.8.4 Data Representation

The AI-201 layer is equipped with a single 18-bit A/D converter. This layer can return either 16-bit 2s complement data in 16-bit words to save on network transmission size, or 18-bit straight binary data in 32-bit words, combined with levels on general-purpose digital I/O lines.

16-bit data is represented as follows:

Bit	Name	Description	Reset State
15	**SIGN**	Sign. Signal levels below 0V correspond with negative values (sign bit is set).	0
14-0	**ADCDATA**	Upper 15 bits of data, twos-complemented	**<pos>**

<pos> represents a position in the output buffer. Upon reset, every entry in the output buffer is filled with its relative position number. As an initializing step, you should read the buffer and discard the data before proceeding with normal data collection.

If you start receiving consecutive data from the layer, like 0,1,2, etc., it means that either the layer is not initialized properly or it is damaged.

To convert data into floating point, use the following formula:

Volts = (Raw ^ 0x8000) * (30V/2^16) – 15V

32-bit data, however, has different representation, as shown below:

Bit	Name	Description	Reset State
31	**ISO_EXT1**	Current value of ISO_EXT1 line (internal use)	0
30	**ISO_EXT0**	Current value of ISO_EXT0 line (internal use)	0
29	**ISO_INT1**	Current value of ISO_INT1 line (internal use)	0
28	**ISO_INT0**	Current value of ISO_INT0 line (internal use)	0
27	**ADCBUSY**	Current value of ADC (1=BUSY) (internal use)	0

Bit	Name	Description	Reset State
26	**DIO2**	Current value of DIO2 (TRIGIN) pin	0
25	**DIO1**	Current value of DIO1 (EXTCLK) pin	0
24	**DIO0**	Current value of DIO1 (CLKOUT) pin	0
23-18	**RSV0**	Reserved, read as 0	0
17-16	**MSB18**	Bits 17 and 16 of the straight binary code in 18-bit mode, 0 in 16-bit mode	0
15-0	**ADCDATA**	16 LSBs in 18-bit mode, twos-complement conversion result in 16-bit mode	0

NOTE: 16-bit mode (18BIT field = 0 in configuration) delivers data in two's complement format[1]; in 18-bit mode, the conversion results are delivered in straight binary 18-bit format.

To convert data into floating point, use the following formula:

Volts = (Raw & 0x3ffff) * (30V/2^18) – 15V

1. To apply two's complement to a base-2 integer, invert the bits and then add one.

Chapter 2 Programming with the High Level API

This section describes how to control the DNx-AI-201 using the UeiDaq Framework High Level API.

UeiDaq Framework is object oriented and its objects can be manipulated in the same manner from different development environments such as Visual C++, Visual Basic or LabVIEW.

The following section focuses on the C++ API, but the concept is the same no matter what programming language you use.

Please refer to the “UeiDaq Framework User Manual” for more information on use of other programming languages.

2.1 Creating a Session

The Session object controls all operations on your PowerDNx device. Therefore, the first task is to create a session object:

```
// create a session object for input
CUeiSession aiSession;
```

2.2 Configuring the Resource String

UeiDaq Framework uses resource strings to select which device, subsystem and channels to use within a session. The resource string syntax is similar to a web URL:

```
<device class>://<IP address>/<Device Id>/<Subsystem><Channel list>
```

For PowerDNA and RACKtangle, the device class is **pdna**.

For example, the following resource string selects analog input lines 0,1,2,3 on device 1 at IP address 192.168.100.2: “pdna://192.168.100.2/Dev1/Ai0:3” as a range, or as a list “pdna://192.168.100.2/Dev1/Ai0,1,2,3”.

2.3 Configuring for Input

The AI-201 can be configured for voltage measurement input. To program the analog input circuitry, configure the channel list using the session’s object method “CreateAIChannel”.

The gain applied on each channel is specified by using low and high input limits.

For example, the DNA-AI-201-100 available gains are 1, 2, 5, 10 and the maximum input range is [-15V, 15V].

To select the gain of 10, you need to specify input limits of [-1.5V, 1.5V]:

```
// Configure channels 0,1 to use gain of 10 in differential mode
aiSession.CreateAIChannel(“pdna://192.168.100.2/Dev0/Ai0,1”,
                          -1.5,
                           1.5,
                          UeiAIChannelInputModeDifferential);
```

Be mindful of your gain setting. Note that when reading any of the channels in point-by-point mode, the hardware actively keeps the data just below the gain-limit. When the gain is set too high, the output will appear as an inverted approximate of the actual signal, scaled down under the gain limit. Try a lower gain value, or begin with one.

2.3.1 Thermocouple Measurement

For thermocouples, use the object method “CreateTCChannel,” which automatically handles temperature calculations, as follows:

```
// Configure channel 0 to 2, scaling for thermocouples,
// thermocouple Type K, degrees F°, using CJC built-in compensation
// from the STP-AI-U board, in differential mode.

aiSession.CreateTCChannel("pdna://192.168.100.2/Dev0/Ai0:2",
          -15.0, 15.0,
          ThermocoupleType.TypeK,
          TemperatureScale.Fahrenheit,
          ColdJunctionCompensationType.BuiltIn,
          0, "",
          AIChannelInputMode.Differential);
```

2.3.2 RTD Measurement

Resistance-Temperature-Detector (RTD) measurements are configured using the Session object method “CreateRTDChannel”.

RTD sensors are resistive sensors whose resistance varies with temperature. Knowing the resistance of an RTD, we can calculate the temperature using the “Callendar Van-Dusen” equations.

RTD sensors are specified using the "alpha" (a) constant. It is also known as the temperature coefficient of resistance, which defines the resistance change factor per degree of temperature change. The RTD type is used to select the proper coefficients A, B and C for the Callendar Van-Dusen equation, which is used to convert resistance measurements to temperature.

To measure the RTD resistance, we need to know the amount of current flowing through it. We can then calculate the resistance by dividing the measured voltage by the known excitation current.

To measure the excitation current, we measure the voltage from a high precision reference resistor whose resistance is known.

The reference resistor is built-into the terminal block if you are using a DNA-STP-AI-U, but you can provide your own external reference resistor, if you prefer.

In addition, you must configure the RTD type and its nominal resistance at 0° Celsius, as shown in the example that follows.

```
// Add 4 channels (0 to 3) to the channel list and configure
// them to measure a temperature between 0.0 and 200.0 deg. C.
// The RTD sensor is connected to the DAQ device using
// two wires, the excitation voltage is 5V, and the reference
// resistor is the 20kOhms resistor built-into the DNA-STP-AI-U.
// The RTD alpha coefficient is 0.00385, the nominal resistance at 0° C
// is 100 Ohms, and the measured temperature will be returned in °C

aiSession.CreateRTDChannel("pdna://192.168.100.2/dev0/Ai0:3",
                0, 1000.0,
                UeiTwoWires,
                5.0,
                UeiRefResistorBuiltIn,
                20000.0,
                UeiRTDType3850,
                100.0,
                UeiTemperatureScaleCelsius,
                UeiAIChannelInputModeDifferential);
```

2.4 Configuring the Timing

You can configure the AI-201 to run in simple mode (point by point) or high-throughput buffered mode (ACB mode), or high-responsiveness (DMAP) mode.

In simple mode, the delay between samples is determined by software on the host computer. In DMAP mode, the delay between samples is determined by the AI-201 on-board clock and data is transferred one scan at a time between PowerDNA and the host PC. In buffered mode, the delay between samples is determined by the AI-201 on-board clock and data is transferred in blocks between PowerDNA and the host PC.

The following sample shows how to configure the simple mode. Please refer to the "UeiDaq Framework User's Manual" to learn how to use other timing modes.

```
// configure timing of input for point-by-point (simple mode)

aiSession.ConfigureTimingForSimpleIO();
```

2.5 Read Data

Reading data is done using *reader* object(s). The following sample code shows how to create a scaled reader object and read samples.

```
// create a reader and link it to the analog-input session's stream

CUeiAnalogScaledReader aiReader(aiSession.GetDataStream());

// the buffer must be big enough to contain one value per channel

double data[2];

// read one scan, where the buffer will contain one value per channel

aiReader.ReadSingleScan(data);
```

2.6 Cleaning-up the Session

The session object will clean itself up when it goes out of scope or when it is destroyed. To reuse the object with a different set of channels or parameters, you can manually clean up the session as follows:

```
// clean up the session
aiSession.CleanUp();
```

Chapter 3 Programming with the Low-level API

The PowerDNA cube and PowerDNR RACKtangle and HalfRACK can be programmed using the low-level API. The low-level API offers direct access to PowerDNA DAQBios protocol and also allows you to access device registers directly.

However, we recommend that, when possible, you use the UeiDaq Framework High-Level API (see **Chapter 2**), because it is easier to use. You should need to use the low-level API only if you are using an operating system other than Windows.

For additional information about low-level programming of the AI-201, please refer to the PowerDNA API Reference Manual document under:

Start » Programs » UEI » PowerDNA » Documentation

Refer to the PowerDNA API Reference Manual on how to use the following low-level functions of AI-201, as well as others related to cube operation:

Function	Description
`DqAdv201Read`	This function works using underlying DqReadAIChannel() but converts data using internal knowledge of input range and gain of every channel.

3.1 Configuration Settings

Configuration setting are passed in `DqCmdSetCfg ()` and `DqAcbInitOps ()` functions.

Not all configuration bits apply to AI-201 layer.

The following bits make sense:

```
// Configuration bits

   #define DQ_FIFO_MODEFIFO (2L << 16)  // continuous
                                        // aquisition with FIFO
   #define DQ_LN_MAPPED     (1L<<15)  // For WRRD (DMAP)devices
   #define DQ_LN_STREAMING  (1L<<14)    // For RDFIFO devices -
                                        // stream the FIFO
                                        // data automatically
                                        // For WRFIFO - do NOT
                                        // send reply to
                                        // WRFIFO unless needed
   #define DQ_LN_IRQEN      (1L<<10)    // enable layer irqs
   #define DQ_LN_PTRIGEDGE1 (1L<<9)     // stop trigger edge MSB
   #define DQ_LN_PTRIGEDGE0 (1L<<8)     // stop trigger edge:
                                        // 00 - software,
                                        // 01 - rising,
                                        // 02 - falling
   #define DQ_LN_STRIGEDGE1 (1L<<7)     // start
                                        // trigger edge is MSB
   #define DQ_LN_STRIGEDGE0 (1L<<6)     // start trigger edge:
                                        // 00 - software,
                                        // 01- rising,
                                        // 02 - falling
   #define DQ_LN_CVCKSRC1   (1L<<5)     // CV clock source MSB
   #define DQ_LN_CVCKSRC0   (1L<<4)     // CV clock source is
                                        // 01 - SW, 10 - HW,
                                        // 11 - EXT
   #define DQ_LN_CLCKSRC1   (1L<<3)     // CL clock source MSB
   #define DQ_LN_CLCKSRC0   (1L<<2)     // CL clock source is
                                        // 01 - SW, 10 - HW,
                                        // 11 - EXT
   #define DQ_LN_ACTIVE     (1L<<1)     // "STS" LED status
   #define DQ_LN_ENABLED    (1L<<0)     // enable operations
```

For streaming operations with hardware clocking, select the following flags:

```
DQ_LN_ENABLE | DQ_LN_CLCKSRC0 | DQ_LN_STREAMING |
DQ_LN_IRQEN  | DQ_LN_ACTIVE
```

`DQ_LN_ENABLE` enables all operations with the layer.

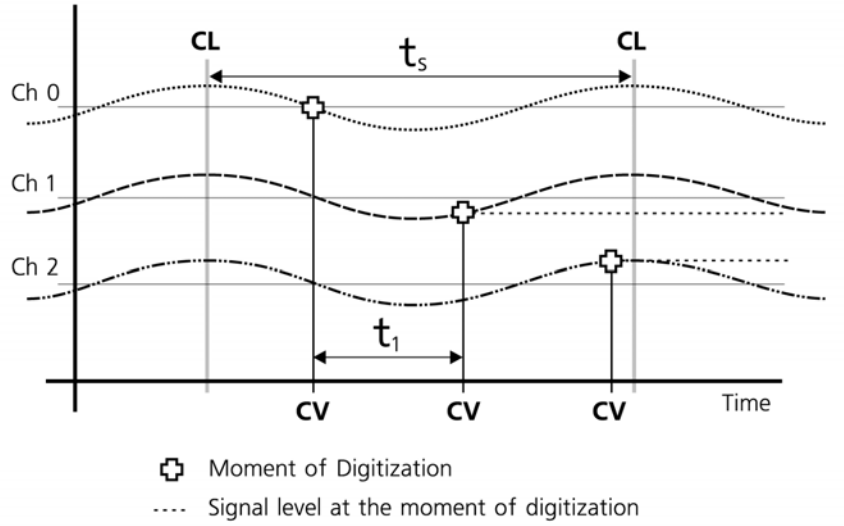
`DQ_LN_CLCKSRC0` selects the internal channel list clock (CL) source as a timebase. AI-201 supports CL clock only where the time between consecutive channel readings is calculated by the rule of maximizing setup time per channel. If you'd like to select the CL clock from an external clock source, such as the SYNCx line, set `DQ_LN_CLCKSRC1` as well.

`DQ_LN_CVCKSRC0` selects the internal conversion clock (CV) source as a timebase. Setting CV clock allows having an equal time period between conversions of different channels. It is mostly used when you are interested in a phase shift between different channels.

`DQ_LN_ACTIVE` is needed to switch on “STS” LED on the CPU layer.

You can select either the CL clock or the CV clock as a timebase. If you select both clocks, the CL clock is taken as a timebase and the CV clock determines the delay between converting channels (i.e. settling time).

In the following picture, CL refers to the CL Clock, also known as the Channel List clock or the Scan Clock. CV refers to the CV Clock, also known as the Conversion Clock.



Note that t_1 shows the time between individual samples on the A/D; the time between CV clock cycles is limited by the board’s maximum digitization rate and settling time. If you need to increase the settling time between samples, slow down the board by decreasing its digitization rate.

Next, t_s is the minimal time between scans of the Channel List; it depends on t_1 and the number of entries in the Channel List. The value of $1/t_s$ is the maximum scan rate in Hz.

The effective per-channel sampling rate also depends on the number of channels in the Channel List. In this case, a layer acquires data across all channels sequentially at the selected speed, which need not be the peak speed, and this rate is called the aggregate rate. When the Channel List contains two channels, for example, the per-channel rate is one-half the aggregate rate. For multiple channels, you can thus calculate the maximum per-channel rate as:

Per-channel rate = Aggregate rate / Number of channels

3.2 Channel List Settings

The AI-201 layer has a very simple channel list structure, as shown in **Table 3-1**.

Table 3-1. Channel List Structure

Bit	Name	Purpose
31	**DQ_LNCL_NEXT**	Tells firmware there is a “next entry” in the channel list.
20	**DQ_LNCL_TSRQ**	Request timestamp as a next data point.
19	**DQ_LNCL_SLOW**	Double the settling time for this channel.
15	**DQ_LNCL_DIFF**	Differential
11..8	**DQ_LNCL_GAIN**()	Gain
7..0		Channel number

Gains are different for various options of the AI-201 layer, as shown in **Table 3-2**.

Table 3-2. Gains

Layer Type	Range	Gain	Gain Number
AI-201-16	±15V	1	0
	±1.5V	10	1
	±150mV	100	2
	±15mV	1000	3
AI-201-100	±15V	1	0
	±7.5V	2	1
	±3V	5	2
	±1.5V	10	3

3.3 Layer-Specific Commands and Parameters

Layer-specific functions for the AI-201 layer are described in DaqLibHL.h file:

DqAdv201Read()

This function works using underlying `DqReadAIChannel ()` but converts data using internal knowledge of input range and gain of every channel. It uses the `DQCMD_IOCTL` command with the `DQIOCTL_CVTCHNL` function under the hood.

When this function is called for the first time, the firmware stops any ongoing operation on the device specified and reprograms it accordingly, with the channel list supplied. This function uses the preprogrammed CL update frequency – 10Hz. You can reprogram the update frequency by calling the `DqCmdSetClk ()` command after the first call to `DqAdv201Read ()`.

Therefore, you cannot call this function when the layer is involved in any streaming or data mapping operations.

If you specify a short timeout delay, this function can time out when called for the first time because it is executed as a pending command and layer programming takes up to 10ms.

Once this function is called, the layer continuously acquires data and every next call function returns the latest acquired data.

If one would like to cancel ongoing sampling, call the same function with 0xFFFFFFFF as a channel number.

Appendix A

A. Accessories

The following cables and STP boards are available for the AI-201 layer.

DNA-CBL-37

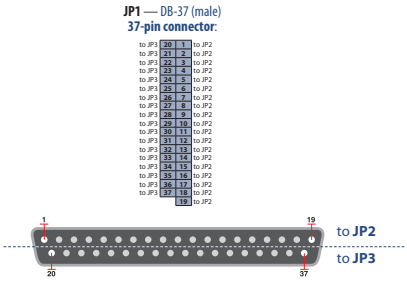
This is a 37-conductor flat ribbon cable with 37-pin male D-sub connectors on both ends. The length is 3ft and the weight is 3.4 ounces or 98 grams.

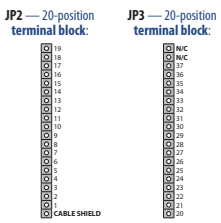
DNA-CBL-37S

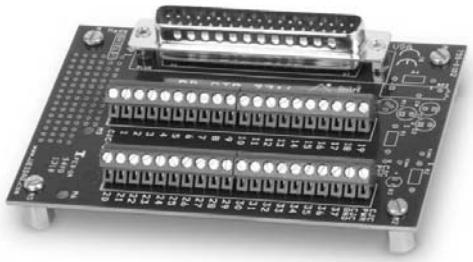
This is a 37-conductor round shielded cable with 37-pin male D-sub connectors on both ends. It is made with round, heavy-shielded cable; 3 ft (90 cm) long, weight of 10 ounces or 282 grams; also available in 10ft and 20ft lengths.

DNA-STP-37

The DNA-STP-37 provides easy screw terminal connections for all DNA and DNR series I/O boards which utilize the 37-pin connector scheme. The DNA-STP-37 is connected to the I/O board via either DNA-CBL-37 or DNA-CBL-37S series cables. The dimensions of the STP-37 board are 4.2w x 2.8d x1.0h inch or 10.6 x 7.1 x 7.6 cm (with standoffs). The weight of the STP-37 board is 2.4 ounces or 69 grams.







DNA-STP-AI-U

Universal screw-terminal panel with embedded CJC.

DNA-STP-AI-207TC

Screw terminal panel for use with the DNx-AI-217 and thermocouples. The panel provides open thermocouple detection as well as the cold-junction compensation measurement.

Index